# SIMULATION STUDY OF SOME TRACKING AND CONTROL METHODS FOR AIRCRAFT INTERCEPT OPERATIONS

CHARLES L. AXELL

SIMULATION STUDY OF SOME
TRACKING AND CONTROL METHODS FOR
AIRCRAFT INTERCEPT OPERATIONS

* * * * *

Charles L. Axell

# SIMULATION STUDY OF SOME
# TRACKING AND CONTROL METHODS FOR
# AIRCRAFT INTERCEPT OPERATIONS

by

Charles L. Axell

Lieutenant Commander, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS

United States Naval Postgraduate School
Monterey, California

1 9 6 4

# SIMULATION STUDY OF SOME TRACKING AND CONTROL METHODS FOR AIRCRAFT INTERCEPT OPERATIONS

by

Charles L. Axell

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

ENGINEERING ELECTRONICS

from the

United States Naval Postgraduate School

ABSTRACT

The use of radar tracking in aircraft intercept operations introduces a random noise factor into the control system. Some techniques have been proposed for organizing and smoothing the radar data prior to insertion in the control system, however the efficiency of these techniques in providing an optimum interpretation of radar data is suspect in the case of maneuvering targets. A simulator design using a general purpose digital computer is proposed to enable observation of the behavior of certain track smoothing methods in a typical intercept problem. The method of aircraft vectoring control in the simulator design is discussed. The effects of angular and linear target acceleration on some tracking methods is described and illustrated. Application of sampled data theory to digital computer programming is shown in a simple filter design. Simulator was programmed by author and the USNPGS CDC 1604 computer was used to conduct testing.

TABLE OF CONTENTS

## LIST OF ILLUSTRATIONS

## TABLE OF SYMBOLS

| Symbol | Meaning | Units |
|---|---|---|
| $\theta_C$ | Interceptor heading | radians |
| $\theta_T$ | Target heading | radians |
| $V_F$ | Interceptor TAS | miles/sec |
| $V_T$ | Target TAS | miles/sec |
| $IAS$ | Indicated Airspeed | miles/sec |
| $\theta_R$ | Target Crossing Angle | radians |
| $\Phi$ | Interceptor Bank Angle | radians |
| $HDG$ | Heading Computed for Intercept | radians |
| $V_{F_R}$ | Interceptor Speed Required | miles/sec |
| $RAD$ | Radius of Turn | miles |
| $ROT$ | Rate of Turn | radians/sec |
| $W_D, W_V$ | Wind Direction, Velocity | radians, miles/sec |
| $ALT$ | Altitude | ft x $10^{-3}$ |
| $X_{T,F}$ $Y_{T,F}$ | Position Coordinates | miles |
| $\tilde{X}$ , $\tilde{Y}$ | Noisy Position Coordinates | miles |
| $\alpha, \beta$ | Tracking parameters | non-dimen. |
| $\bar{X}$ , $\bar{Y}$ | Smoothed position | miles |
| $\bar{\dot{X}}$ , $\bar{\dot{Y}}$ | Smoothed velocity | miles/sec |
| $\hat{X}$ , $\hat{Y}$ | Predicted position | miles |

## 1. Introduction

The use of radar as a data gathering device has gradually been supplemented by the introduction of processing and computing equipment designed to make the most efficient interpretation of the information thusly obtained. One factor that complicates the process of interpretation is the presence of random error in the data which results from the resolution qualities of the radar. The problem of filtering this data has been the subject of many papers since that presented by Wiener two decades ago. [1] In some applications, however, the theory presented cannot be directly used. One such situation is that which remains a primary problem in the air intercept control system; the tracking of a maneuvering aircraft. Several methods of tracking and smoothing have been advanced for use in this system which purportedly provide sufficiently smoothed data for intercept geometry calculations. Whether or not these methods, using weighting function response techniques over a finite time, can provide data of sufficient smoothness and accuracy is a question which can be answered only by controlled testing in a typical system. The most practical way of conducting such tests is through use of a simulator which will duplicate the operation of the entire air intercept control system.

The purpose of this paper is to report on the design and operation of such a simulator, programmed into a general purpose digital computer, and to investigate the results of some limited tests conducted thereon. The programming methods and subroutines used in constructing the simulator were chosen with little regard for arithmetic efficiency or mini-

mizing storage space, and hence are not submitted as system proposals. Rather they are presented as being representative of methods which might be used in such a system.

## 2. Preliminary Analysis

The data obtained from a radar set provides range, azimuth and elevation signals in discrete, sampled form, with sampling rate of 1/T where T is the period of antenna rotation. A video processor operating on this output would ideally remove all spurious data and pass on target information with a blip to scan ratio of one, still in discrete form. For simplicity it will be assumed that the miss probability is zero. This sampled data, which includes the target position signal plus a random error function in range and azimuth due to a position uncertainty inherent in the radar, must then be operated on by a position and velocity smoothing procedure to obtain a best estimate of track of each target. Once the tracks are established, computations are made based on desired tactics so as to control the interceptor aircraft into a particular position relative to the target. The maneuvering of the interceptor in response to the control commands is interpreted by the radar and the system is then closed except for the inclusion of human decision factors.

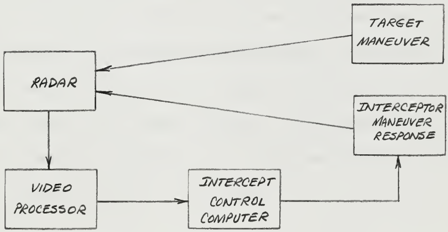



Figure 1. Block diagram of Simulator System

The simulation of this closed system is accomplished using a digital computer (CDC 1604) with FORTRAN chosen as the programming language. The program runs in fast time while maintaining a real time index for scaling velocities and turn rates. Of the two possible methods of control computations recognized, that of performing a complete solution at the onset and updating as necessary, and that of making a partial solution with each sampling, the latter was chosen to permit overall use of sampled data theory without the need of adjusting for a variable time base. A flow diagram of the simulator design used is included as Appendix I.

A. Intercept Control Computer Program.

The method by which the computer generates control information is not considered to be a critical factor in the simulation of the system. However, it is felt that a general knowledge of the method used will assist the reader in following developments made later.

The geometry of the intercept was chosen to be one in which the interceptor is vectored for a collision course intercept with an offset point which is a function of target crossing angle, velocities of target and interceptor, and a particular bank angle to be used for the turn from the offset point to the target heading. This offset point is calculated so as to put the interceptor at a point one half mile astern of the target on completion of the final turn.

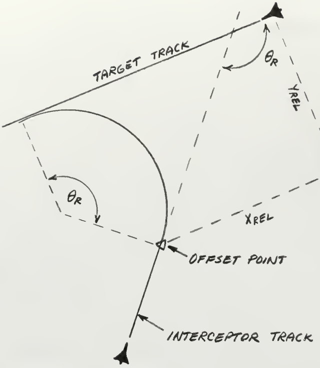



Figure 2. Offset Point Geometry

In Figure 2, the X and Y relative distances are computed using the formulas:

$$X_{REL} = V_T \frac{|\theta_R|}{ROT_F} - .5 - RAD_F \sin(|\theta_R|)$$

$$Y_{REL} = RAD_T (1 - \cos(|\theta_R|))$$

$$SIDE = \frac{\theta_R}{|\theta_R|}$$

and, by flagging the side from which the intercept is to be made, these relative distances are rotated to the x, y reference by

$$X_{ABS} = X_{REL} \cos\theta_T - SIDE \cdot Y_{REL} \sin\theta_T$$

$$Y_{ABS} = X_{REL} \sin\theta_T + SIDE \cdot Y_{REL} \cos\theta_T$$

During the phase of the intercept where this point is being intercepted, the offset coordinates are added to the target coordinates to generate a pseudo-target on which the collision course calculations are made.

The calculations for the collision course intercept are based on relationships in Figure 3,

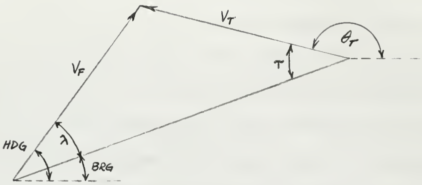



Figure 3. Intercept Geometry

where the bearing to target is $BRG = \tan^{-1}\left[(Y_T - Y_F)/(X_T - X_F)\right]$

and the relationships

$$\tau + \theta_T = \pi + BRG$$

$$\tau = \pi + BRG - \theta_T$$

are used in the sine law to find HDG solution

$$\lambda = \sin^{-1}\left[\frac{V_T \sin\tau}{V_F}\right]$$

and intercept heading then becomes $HDG = BRG + \lambda$.

No mention of wind is made in these calculations. Since wind at normal interception altitude may be as high as 25% of the aircraft speeds, considerable difference will exist between actual track fed in by the video processor and computer output commands. Also, target heading and true air speed will not be coincident with track parameters and will cause errors in determining a precise intercept course and offset point. Although it would be impossible to predict winds throughout the inter-

cept progress for complete compensation of these effects, a good estimate could be used based on prior knowledge which would reduce these effects to an insignificant size. In the program used, there is a routine inserted to convert the track smoothing output into a no-wind situation prior to insertion into the computer program, thereby eliminating the necessity of including wind in the actual intercept computations.

The intercept, on reaching the offset point (within distance limits), then proceeds with a constant rate turn of $\Phi = 30^{\circ}$ toward the target heading. To realistically complete this turn, when the relative target bearing reaches a computed antenna train angle, the turn changes into a target tracking course which simulates the conversion to aim dot steering from the airborne computer system.

The output of the intercept control portion can either be continuous, as in the case where data link communications exist between computer and interceptor, or of the step output form, where commands are passed only on significant change of the computer solution, which duplicates the present manual control method. The former method involves the generating of a command acceleration (linear and angular), and the latter the setting of a heading and altitude. The latter is used in this program, and the method selected for generating the command headings and altitudes will be investigated later.

## B. Interceptor Response Simulation

The reaction of the simulated interceptor to commands from the control computer section is simultaneous. Although pilot reaction time is not negligible, and would in practice be accounted for, it is assumed

to be compensated for in this simulator. Hence, on command for a turn to a new vector, the simulator sets the turn rate corresponding to the present true airspeed using the aerodynamics relationships,

$$RAD = 189.3\ (TAS)^2/\tan\Phi$$

$$ROT = (0.00529\ \tan\Phi)/TAS$$

where PHI = bank angle in radians

TAS = true airspeed in miles/sec

ROT = rate of turn in radians/sec

RAD = radius of turn in miles

The turn rate is then converted to radians/time base, and the x, y coordinates of the interceptor are modified according to the relationships shown in Figure 4.

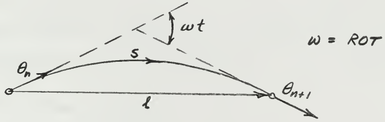



Figure 4. Turn Compensation

where $\theta_{n+1} = \theta_n + \omega t$ , ( $t$ = radar scan time )

where the arc length is the actual distance travelled,

$$s = RAD\ (\omega t)$$

and the chord length,

$$l = 2\ RAD\ \sin\left(\frac{\omega t}{2}\right)$$

are related by the factor

$$CONV = \frac{l}{s} = \frac{\sin\left(\frac{\omega t}{2}\right)}{\frac{\omega t}{2}} = \left[ 1 - \frac{\omega^2}{24} + \cdots\cdots \right]$$

and advance of the coordinates are made by the formulas

$$X_{F_{n+1}} = X_{F_n} + \left[V_F\,(CONV)\cos\left(\theta_F + \frac{\omega t}{2}\right) + W_X\right] t$$
$$Y_{F_{n+1}} = Y_{F_n} + \left[V_F\,(CONV)\sin\left(\theta_F + \frac{\omega t}{2}\right) + W_Y\right] t$$

where WX and WY are wind drift inc ements. The target is advanced in the same manner according to a programmed maneuver. Headings are advanced according to turn rate,

$$\theta_{F_{n+1}} = \theta_{F_n} + \omega_F t$$
$$\theta_{T_{n+1}} = \theta_{T_n} + \omega_T t$$

and when the heading comes within a present range of the new vector, a random deviate is looked up and the interceptor course is set within $\pm 5^{\circ}$ of this vector (normal distribution) to simulate error in heading control.

Speed and altitude control responses are simply straight line variations, with speed change set with a constant acceleration of ± .00028 miles/sec$^2$ , and altitude change based on the need for reaching the new altitude on expiration of 50% of the time to go until intercept.

C. Radar and Video Processor

The 'actual' blip positions representing true target positions are taken from the interceptor response simulator in rectangular coordinates and converted to polar form. According to the resolution capabilities of the radar set being simulated, a Gaussian noise is added to the range and bearing signals, which are then translated back into rectangular form. The processor output is set at a blip to scan ratio of one by this procedure, and the hit probability is assumed to be one.

## 3. Systems Design

### A. Control Methods

The intercept computer generates a solution based on present geometry with each iteration. Since it is assumed that this simulator is to represent a system which is limited by the computer-interceptor communications link to a step command output, a non-linear system must be considered in generating this output. The simplest system is one wherein a new heading command is given at the time the computer solution differs by a set amount from the last command, the new heading command taking the value of the solution at that time. If the interceptor were able to respond instantly to this command, and hold the exact heading ordered, with no time variable parameters involved, this one order would be correct during that particular phase of the intercept. Such is not the case, however, since the response time of the interceptor is proportional to heading change and turn rate, and since parameters are involved which are time variable, such as linear acceleration to intercept speed and target heading during track smoothing. Also, there is an inherent steering error and speed control error on the part of the interceptor which is not predictable and which along with erroneous wind estimates causes a drift in the computer solution. Hence, by the time the interceptor reaches the new heading, it is no longer the correct heading, and the result is a sucession of heading commands lagging the correct value.

One of the operational factors involved in controlling the interceptor is the necessity of reducing the number of control commands to a minimum, to avoid communications saturation. To accomplish this end, it

is necessary to anticipate interceptor response and computer solution behavior in determining the next command to be given, in order that a more nearly correct heading will result (the event in which the interceptor will take the exact heading required has a probability of zero). A simple method of prediction is that of using the derivative (continuous case) of the required heading and assuming a linear change during the time the interceptor completes the rurn, as in Figure 5.

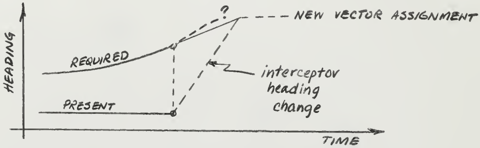



Figure 5. Sketch of Derivative Prediction

The error involved in this method depends on the amount of nonlinearity present, which is indicated by the order of the polynomial required to fit a curve to sampled points. Using data from an early trial run it was found that a good approximation could be obtained by a third order polynomial at short ranges, and a second order polynomial where the range to target was over 50 miles. This would indicate that the simple constant plus derivative (difference) method might deteriorate at short range where heading control is critical.

Another possible predictor, of greater complexity, would be an extrapolator using a difference table of some chosen order. In order to compare this method with the previous to determine if any significant advantage might be obtained that would outweigh the computer time involved, both methods were synthesized in the simulator. The main ad-

vantage in finding an efficient and accurate predictor lies in the resultant minimization of steering error afforded by providing the most correct mean heading about which the steering error is applied.

To predict the required HDG at some future time based on past history, a third order polynomial is fitted to the last four HDG solutions, using the divided difference table.

$$\begin{array}{cccccc}
\underline{t} & \underline{f(t)} & & & \\
t_0 & f_0 & & & \\
 & & \Delta f_0 & & \\
t_1 & f_1 & & \Delta^2 f_0 & \\
 & & \Delta f_1 & & \Delta^3 f_0 \\
t_2 & f_2 & & \Delta^2 f_1 & \\
 & & \Delta f_2 & & \\
t_3 & f_3 & & &
\end{array}$$

Although the actual function is of higher order, the error in assuming a third order fit is small compared to steering errors. Hence it will be assumed that

$$\Delta^3 f_1 = \Delta^3 f_0$$

Now, rather than using Newton's forward interpolation formula, an extrapolation is made based on last differences,

$$\Delta^2 f_2 = \Delta^2 f_1 + \Delta^3 f_0 \quad ; \quad \Delta f_3 = \Delta f_2 + \Delta^2 f_2$$

In order to make this system adaptable to varying time base, the special formulas are used,

$$\Delta^2 f_2 = \Delta^2 f_1 + (t_n - t_1)\Delta^3 f_0 \ ; \ \Delta f_3 = \Delta f_2 + (t_n - t_2)\Delta^2 f_2$$

and to find $f_n$ at some given time following $t_3$,

$$f_n = f_3 + (t_n - t_3)\left\{\Delta f_2 + (t_n - t_2)\left[\Delta^2 f_1 + (t_n - t_1)\Delta^3 f_0\right]\right\}$$

where $t_n$ is solved for using

$$\Delta f_2 \doteq \left.\frac{d(f(t))}{dt}\right|_{t=t_3} \quad ; \quad t_n = t_3 + \frac{(f_3 - \text{PRESENT HEADING})}{(ROT_F - \Delta f_2)}$$

according to the sketch shown in Figure 6.

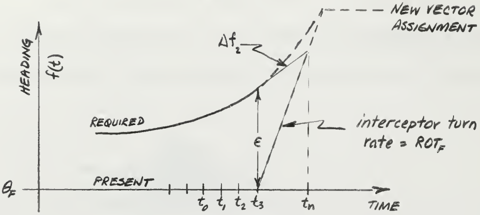



Figure 6. Sketch of Extrapolation using Last Differences

On initial vector assignment, extrapolation may cover a very long period and involve a significant error, however, since the solution is made continuously throughout the resultant turn, it is self-correcting.

B. Tracking and Smoothing Procedures

The output of the video processor consists of sampled signal plus noise information, in the sense that noise exists as a random error in position, with predictable variance and an assumed Gaussian distribution. In the translation of these polar form data into rectangular coordinates, the independent random variables associated with range and azimuth error become functions of position. Assuming that range and azimuth signals have the respective deviations $\sigma_\rho$ and $\sigma_\theta$

then in converting to the rectangular form, the variance associated with the x and y coordinates become

$$\sigma_x^2 = \sigma_\rho^2 \cos^2\theta + \sigma_\theta^2 \rho^2 \sin^2\theta$$

and

$$\sigma_y^2 = \sigma_\rho^2 \sin^2\theta + \sigma_\theta^2 \rho^2 \cos^2\theta$$

Graphically, the joint probability distribution of the random variables in polar coordinates at a particular point would appear as in the figure below,

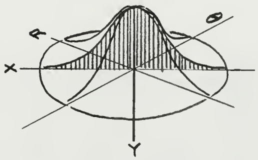



Figure 7. Sketch of Joint Prob. Distribution of Typical Radar Report

with the transformed x-axis random variable distribution shown by the cross-section. On successive scans of the radar, with the processor providing a blip to scan ratio of one, a set of random variables is produced whose correlation is dependent on the shape of the actual track. Given a sample of sufficient size, a regression curve could be fit based on measurements obtained using statistical analysis. In the event that the dependence of x and y were linear, as in a straight track, the slope and y-intercept could be found directly. However, this method is not satisfactory since the information required is available only after future samples are taken, and since new position information is not directly available. A more desirable means of estimating the linear re-

lationship of the x and y coordinates would be a recurrent updating of a track estimate based on previous values using a weighing function according to track length.

(1) Constant Parameter Track Smoothing

Some of the early work done in the field of track position and velocity smoothing is that presented by Boxer in extention of the formulas of S. Thaler [2]. He uses the difference equation representation of a digital computer operation to construct and analyze a feedback network representing a positional tracker with velocity aiding. In the block diagram,

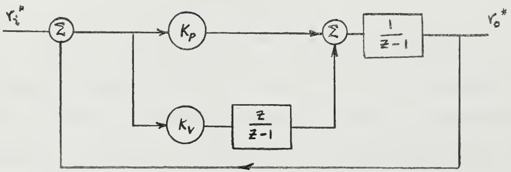



Figure 8. Block Diagram of Velocity Aided Positional Tracker

the $\frac{1}{z-1}$ term represents a digital accumulator and the z in the numerator of the other term provides output without delay. The overall transfer characteristics of this system can be determined by z transform methods, yielding in recurrence form:

$$r_o^*(t+2T) + \alpha r_o^*(t+T) - \beta r_o^*(t) = (K_p + K_v)\Delta r_i^*(t) + K_v r_i^*(t)$$

where

$$\Delta r_i^*(t) = \frac{r_i^*(t+T) - r_i^*(t)}{T} \quad , \quad \beta = K_p - 1 \quad , \quad \alpha = K_p + K_v - 2 \, , \quad T = 1$$

Letting the driving function $r_i^*(t)$ be equal to zero provides the characteristic equation, whose Laplace transform is

$$R_o^*(z)\left[z^2 + \alpha z - \beta\right] = 0$$

The stability criterion here is that $\alpha$ and $\beta$, the roots of $z^2 + \alpha z - \beta = 0$, have absolute values less than unity. This describes the allowable values for $K_p$ and $K_v$ according to the figure

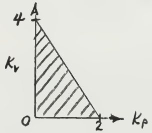



Figure 9. Stability Area for Smoothing Parameters

to lie within the shaded portion for stability. Further analysis of stability is advanced by Boxer in forming the over-all transfer function of the system and taking the denominator as an equivalent Nyquist equation. The same results obtain.

By labeling the signals as shown in the figure below,

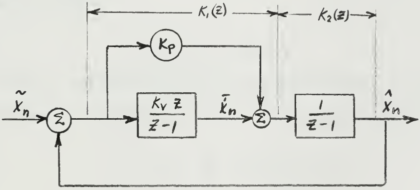



Figure 10. Block Diagram of Velocity Aided Tracker

we can develop the relationships as follows:

$$K_1(z) = \frac{O_1(z)}{I_1(z)} = K_v\left(\frac{1}{1-z^{-1}}\right)$$

from which we get

$$O_1(z) = K_v I_1(z) + O_1(z) z^{-1}$$

or,

$$O(t) = K_v I(t) + O(t-T)$$

$$\bar{\dot{x}}_n = K_v(\tilde{x}_n - \hat{x}_n) + \bar{\dot{x}}_{n-1}$$

where the input is taken to be $(\tilde{x}_n - \hat{x}_n)$. The input to the second accumulator is then

$$\bar{\dot{x}}_n + K_p(\tilde{x}_n - \hat{x}_n)$$

and the transfer function

$$K_2(z) = \left(\frac{z^{-1}}{1-z^{-1}}\right) \quad ; \quad O_2(t) = I_2(t-T) + O_2(t-T)$$

gives the output as

$$\hat{x}_n = \bar{\dot{x}}_{n-1} + K_p(\tilde{x}_{n-1} - \hat{x}_{n-1}) + \hat{x}_{n-1}$$

If we define the quantity

$$\bar{x}_{n-1} = \hat{x}_{n-1} + K_p(\tilde{x}_{n-1} - \hat{x}_{n-1})$$

then we can write

$$\hat{x}_n = \hat{x}_{n-1} + \bar{\dot{x}}_{n-1}$$

Thus showing that this development leads directly to the prediction and update formulas generally used for track position and velocity smoothing.

$$\left\{\begin{aligned} \bar{x}_n &= \hat{x}_n + \alpha(\tilde{x}_n - \hat{x}_n) && (1) \\ \bar{\dot{x}}_n &= \bar{\dot{x}}_{n-1} + \beta(\tilde{x}_n - \hat{x}_n) && (2) \\ \hat{x}_{n+1} &= \bar{x}_n + \bar{\dot{x}}_n \end{aligned}\right\} \ldots \text{ for } T = 1$$

where the substitutions $\alpha = K_p$ and $\beta = K_v$ have been made.

The preceeding plot showing the stability limits for $\alpha$ and $\beta$ indicates that there is no unique set of values, and the possibility is that the performance of the system may be adjusted for optimum response to a particular set of input values by varying these parameters.

(2) Variable Parameter

By defining a variance in the input signal, and using the least squares method of fitting a straight line to the input, FCPC has developed recurrence formulas to be used in a computer to control the $\alpha$ , $\beta$ parameters [3]. Such a method, when used with an appropriate turn logic, provides optimal smoothing during those portions of the tracking problem where no turns occur. The derivation of these formulas follows from the definition that the best value of a set of measurements is that for which the sum of the variances of the observed values is a minimum. Thus from (1),

$$\bar{X}_n = \alpha_n \tilde{X}_n + (1-\alpha_n)\hat{X}_n$$

we define the variance of $\bar{X}_n$ to be the square of the constants of the independent terms $\tilde{X}_n$ and $\hat{X}_n$ times their respective variances,

$$\bar{V}_n = \alpha_n^2 \tilde{V}_n + (1-\alpha_n)^2 \hat{V}_n$$

and from (2),

$$\bar{\dot{X}}_n = \beta_n \tilde{X}_n - \beta_n \hat{X}_n + \bar{\dot{X}}_{n-1}$$

using same definition for variance of $\bar{\dot{X}}_n$ :

$$\bar{\dot{V}}_n = \beta_n^2 \tilde{V}_n + \beta_n^2 \bar{V}_{n-1} + (1-\beta_n)^2 \bar{V}_{n-1} - 2\beta_n(1-\beta_n)COV_{n-1}$$

where covariance COV is the expected value of the product of the deviations

of the variables from their expected values.

The FCPC report develops a set of final equations based on this type of analysis in which the values of $\alpha$ and $\beta$ are computed each pass according to equations that would minimize $\overline{V}_n$ and $\overline{\dot{V}}_n$ respectively for a constant velocity input variable [3]. These equations are not repeated here due to the classification of the reference, however, they are used in the system and identified as the variable parameter method.

C. Data Filtering

Once the program determines smoothed values for target position and component velocities, $\overline{X}$, $\overline{Y}$, $\overline{\dot{X}}$, $\overline{\dot{Y}}$, the track is calculated and ground speed computed directly from $\overline{\dot{X}}$ and $\overline{\dot{Y}}$. These are then fed into the computer for intercept calculations. It was noted that small errors in heading and track has a multiplicative effect in the offset point calculations, causing significant excursions from the expected values. By comparing the offset point track in Figures 13 (noiseless) and 16, it can be seen that this effect is detrimental, causing excessive hunt for heading on the part of the interceptor. It was thought that a special smoothing technique in the computing program itself, in the form of a sub-routine, would be beneficial to stabilize the calculations.

Using the relationships which exist between the sampled data system and the digital computer, and with the program arranged to iterate each radar sweep, it is possible to apply sampled data system theory directly to the problem. Hence, we can construct a low pass filter, using the analogy of a friction damped mass in a simple position feedback network. Such a system can be drawn as in Figure 11 as a continuous system.

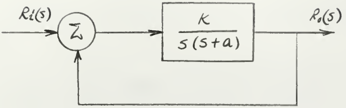



Figure 11. Simple First Order Positioning System

Its sampled data equivalent is shown in Figure 12.

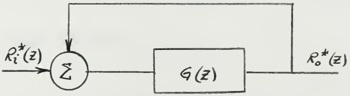



Figure 12. Sampled Data Equivalent of Figure 11

where the $a$ and $K$ can be chosen to obtain the desired 'filter' characteristics.

Since the noise in the system is predominantly of the frequency $\frac{1}{2T}$, the cut-off frequency of the filter should be well below this value. Writing the continuous signal over-all transfer function

$$W(s) = \frac{R_o}{R_i}(s) = \frac{K}{s^2 + as + K}$$

and arbitrarily choosing a damping ratio of about .7 and a bandwidth of about $\frac{\pi}{4T}$ radians/sec.

$$K = \omega_n^2 = \left(\frac{\pi}{4T}\right)^2 = (0.196)^2 = 0.0375 \qquad (T = 4)$$

and,

$$a = 2\zeta\omega_n = 0.275$$

with

$$aT = 1.1 \quad ; \quad \omega_n = 0.196$$

Since overall transfer contains no finite zeroes, bandwidth is given by relationship,

$$B.W. = \omega_n \left[ 1 - 2\zeta^2 + \sqrt{2 - 4\zeta^2 + 4\zeta^4} \right]^{\frac{1}{2}}$$

which in this case is actually $\omega_n$ . Now switching from Laplace to z-transform, the open loop transfer is

$$\mathcal{Z}^{-1}\left[G(s)\right] = \mathcal{Z}^{-1}\left[ \frac{0.14}{s} - \frac{0.14}{s+.275} \right] = \frac{0.0466\, z^{-1}}{(1 - z^{-1})(1 - .3333\, z^{-1})}$$

making the overall transfer function

$$\frac{C}{R}(z) = \frac{0.0466\, z^{-1}}{1 - 1.2867\, z^{-1} + 0.3333\, z^{-2}}$$

Taking the inverse transform of this equation and putting it into the time solution,

$$C(t) = 0.0466\, R(t-T) + 1.2867\, C(t-T) - .3333\, C(t-2T)$$

produces a recurrence formula which uses values associated with the two previous sampling times in computing output. A delay of $T$ seconds is inherent, and response to various input signal functions is dependent on the parameters chosen. It is anticipated that the signal to be filtered contains no step function components and that the size of steady state error to ramp and acceleration inputs is not critical. For the values chosen, typical response to a noisy signal with step components is shown in Figures 17 and 18.

The linking of sampled data system analysis with digital programming in this manner allows the full use of response, compensation, and stabi-

lity theorems in designing the program routine. An important fact to consider, however, is that while the system transfer function is arbitrarily selected, the resultant poles and zeroes are exact and stationary, enabling complete cancellation if that form of digital compensation is envisioned. Hence, the filtering action of the system would be lost. Also, the use of derivative feedback, which merely adjusts the position of the control poles, is likewise meaningless since we are free to choose at the outset which poles we desire. There remains then the theorems governing response according to the input anticipated. Since this input will likely include step, ramp and acceleration inputs of some magnitude, it seems most desirable to adjust the system for a minimum overshoot response to a step input while holding steady state error to a ramp input within reasonable limits. Also, it would be possible to set initial error at zero with the first step input and to jump the output to this value, thus eliminating the initial step response problem. The simulator allows us to examine the type input signal generated, and to choose the most efficient method of design. The use of the initializing routine at the start of the run eliminates initial overshoot problems attendant with the first step input discontinuity, allowing concentration in the filter design of maximum smoothing consistent with allowable steady state error to ramp inputs. Redesigning the filter with this objective, it is possible to achieve better smoothing without being penalized by sluggish response to step inputs. It should be noted that the object of this filter is primarily to remove random 'error' signals from system variables in order to stabilize the calculations for the offset distances, and that small steady state errors are quite tolerable in

view of the magnitude of steering and speed errors already present in the computations. Were this not so, then closer attention would have to be paid in selecting the velocity and acceleration constants of the filter system.

An alternate filter design using a damping ratio of $.9$ and $\omega_n = .2$ was used, with step-sense included, having the overall transfer function

$$\frac{C}{R}(z) = \frac{0.0848\, z^{-1}}{1 - 1.1517\, z^{-1} + .2365\, z^{-2}}$$

with a corresponding recurrence formula of

$$C(t) = .0848\, R(t-T) + 1.1517\, C(t-T) - .2365\, C(t-2T)$$

The logic used for the step-sense, along with the representative recurrence iteration, appear in FORTRAN language as:

```
      IF (K - 1)  99, 99, 100
   99 R1  =  Variable
      C1  =  R1
      C2  =  C1
  100 C   =  A1 * R1  +  A2  *  C1  +  A3  *  C2
      C2  =  C1
      C1  =  C
      R1  =  (variable)
  101 CONTINUE
```

where A1, A2, and A3 are preset constants.

## 4. Simulator Testing and Analysis

Three simulator configurations were used in conducting the tests of the system. All were programmed with the same geographical positioning of the target and interceptor as shown on Figure 13, with headings and speeds and altitudes as indicated. In each, the interceptor was programmed to make a beam attack from the nearest side. The programs differ in that system I has a 'noise' free input, providing exact track information to the computer. System II has a constant parameter tracker-smoother, and system III has a variable parameter tracker based on least-squares analysis.

### A. Control Methods

Simulator configuration I was intended to allow analysis of the control methods outlined in section 3 The results of runs made of this configuration are shown in Figures 13 - 15. As the labelling indicates, in Figures 14 and 15, the simple derivative predictor, when used for controlling the interceptor heading, is not able to anticipate the rate of heading change that actually exists, and ends up following the input at shorter ranges to the extent that an appreciable lag may exist at the critical turn point. If the aircraft speeds were increased, the input heading drift rate would also increase, causing a marginal situation in completing the intercept with the accuracy desired. The extrapolator does a better job in making these predictions, and in the problem run, resulted in fewer heading command outputs. It should be noted that in the testing of these methods, there was a random error factor inserted in the interceptor response to each heading command, of the

range shown in the figures, so that the results would be more directly applicable to the actual problem. In this light, we should also have to consider the track accuracy required in the practical problem. Since this phase of the intercept brings the interceptor to a point from which a final rendezvous type turn is made, during which time compensation can be made for a fairly large positioning error, there could be a wide latitude for error without affecting the final success of the intercept. In this analysis, however, the major concern is to remedy known sources of error so as to keep the cumulative error to a minimum.

B. Simulated Tracking and Smoothing

The basic organization of the intercept control computer is complex in the switching functions and decision routines that are necessary to complete a simple intercept. The equations themselves are straightforward, however, and the construction of the simulator becomes a problem in programming. The operation of this control system when linked with a noise-free feedback path of position and velocity as shown in Figure 13 is smooth and predictable, due to the linear behavior of the variables involved in the intercept computations. Such is not the case when the feedback path is impressed with random noise. Now the input variables to the control unit are non-linear, and this effect is magnified in the computer solutions, resulting in the type of intercept shown in Figure 16. The random movements of the offset point track demonstrates this effect. In this intercept, the fixed parameter method of track smoothing was used with a turn logic shown in Appendix II. The turn logic is designed to sense a departure of an incremental heading from the previous smoothed value by a certain set margin, at which time

an initiation of a new track is signalled. It does not sense a departure of incremental velocity, however, and this appears to be a serious drawback in using this turn (only) logic. In Figure 17 the response is shown of this tracker to a ramp velocity function corresponding to the acceleration of the interceptor to attack speed. Since the interceptor speed requirement is interpreted from the target speed, which is derived in the same fashion, the target maneuver in the vicinity of the 65th iteration causes multiplicative effects in the program. By analysis of the results shown in Figure 17 it is apparent that this tracking technique cannot be used directly throughout a turn maneuver, even by initializing with each scan, in view of the error excursions generated each time a turn occurs. This is due in part to the error involved in using the chord length throughout a portion of the turn as being the actual distance travelled. This error is equal to the departure of the $\frac{\sin x}{x}$ curve from unity for the angle of turn during which the approximation is applied. A much more significant error is also present, prior to the conversion of the $x$ and $y$ velocities into heading and velocity. As the turn progresses through the heading of one of the major axes, the velocity associated with the other axis changes sign. In the recurrence or weighting function, no provision is made to accomodate this rapid a change of velocity, and hence the velocity lags while at the same time the velocity associated with the other axis remains fairly constant, producing a significant error during conversion to resultant heading and velocity. On actuation of the turn logic, and initializing of the tracking routine, this error is temporarily resolved. The critical value appears to be the limit set in the turn logic which determines when an

actual turn has begun. If this limit is too small, input signals with large deviations would trigger the logic and upset otherwise good smoothing. If it is too large, an excessive amount of turn would occur before triggering the logic, resulting in large errors in speed and heading during the turn. In the tests made, this variable was set to sense when any incremental variation in heading differed by more than 11½ degrees from the smoothed value, and was not changed during any of the runs made. It is also apparent from the figure that the tracker fails to follow a linear acceleration, which is also due to the constant and relatively small weighing factor $\beta$ that serves to update the velocity. The dependence of the tracker routine on the turn logic chosen is evident in Figures 21 and 22 where the radar resolution was purposely deteriorated to increase the noise in the system. It can be seen that the output of the tracker more closely follows target maneuvers in heading and speed, at the expense of providing poor smoothing during these maneuvers. This resulted from the more rapid triggering of the turn logic by the input signals of larger deviation.

While a re-design of the tracker and turn logic equations is certainly indicated, the probability is that the random nature of the output will not be completely reduced. To investigate the possibility of using additional smoothing in the control program, the filter described in section 3C was inserted to smooth selected input values. The degree of smoothing provided in the case of interceptor velocity is shown in Figure 17, and in the improved overall solution as shown in Figure 19, where filtering was performed on interceptor velocity, target velocity, and target crossing angle.

Of interest is the fact that the hypothetical radar selected for these tests had relatively superior resolution capabilities of about 1/8 mile and 15 minutes. A reduction in this performance to a resolution of 0.3 miles and 35 minutes, using the same scan rate resulted in the solution shown in Figure 20, with filter performance shown in Figures 21 and 22.

The third simulator configuration used the tracking and smoothing procedures based on least squares fit as outlined in section 3B(2). Again, this method is intended for straight line tracking and required a turn logic routine. This routine is shown in Appendix III. In the testing of this configuration, it appeared that a separate logic was needed to sense linear acceleration as well as angular, since the variable weighing function associated with velocity would not respond to an increase in target speed rapidly enough, as seen in Figure 18. Since the performance of the tracking methods used seems to be largely dependent on the choice of turn logic used, an evaluation of these methods cannot be made on a maneuvering target, and no conclusions can be drawn as to the merits of these methods in the intercept control system.

If the tracking systems were considered as being representative of some sampled data system, such as the analogy shown in the constant parameter system, with the recurrence formulas representing the weighting function obtained by the inverse transform of the transfer function, then we could assume the system input to be a noisy ramp input (of position) and a noisy step input of velocity. Initialization of the routines corresponds to setting initial error at zero, and the response of the system is designed to produce minimum error for the expected input (con-

stant velocity). Now if we assume that the position input will be of second order, as is the case with a maneuvering target, it is necessary to redesign the system for minimum response error to this type signal also. Then, realizing the dependence of the $x$ and $y$ coordinates of a realistic target, we should also solve for heading and velocity and perform identical smoothing on these variables. With such a system, a separate turn logic would not be necessary. Since the design of such a routine is not the intent of this paper, the foregoing is submitted merely as a proposed approach based on the observances made thus far using the simulator routine.

## 5. Conclusions

The use of a general purpose digital computer in simulating an air intercept control problem has enabled the author to investigate the behavior of a hypothetical yet typical control system and two proposed tracking and smoothing procedures. The results of the limited investigations made are valuable only in so far as the arbitrarily selected environmental conditions are concerned. It was found that the constant parameter smoothing method relies heavily on the appropriate choice of turn logic in producing a heading output of any instantaneous accuracy throughout a turn maneuver, and that velocity output throughout this maneuver is quite erratic. The choice of this turn logic appears to hinge on the scan rate and the anticipated rate and duration of turn, rendering this elementary tracking method inefficient for use in the system.

The variable parameter method proved to be rapidly convergent to a very smooth heading and velocity output during straight line portions of the track being followed. However, like the constant parameter method, it depended heavily on an appropriate turn logic for even the most approximate outputs during a turn maneuver. The basic problem in both methods was the inability of the tracker to follow component velocity changes throughout a turn or throughout linear acceleration. A turn logic which would function in a manner similar to the tracker in weighting a turn rate variable and linear acceleration variable appears to be the next logical extention of these methods, and is suggested as a possible area of further study.

The requirements of the control system in computing a satisfactory solution to the intercept problem were found to be quite flexible. The

most detrimental effects were seen to originate more from noisy input variables than from the erroneous but smoothed, filtered form of these variables for the type intercept solution made. If the offset point method were to be used with an additional beam displacement, as is most probably the case in present air tactics, then the effects of noisy variables would be more acute and would accentuate the need for additional filtering of the heading and speed variables.

It is hoped that this paper will provide a partial ground-work in the intercept problem from which further investigation may be made of some of the problem areas revealed. It is suggested that the program shown in Appendix I be oriented about a method of simultaneous display and direct computer control in order to allow more rapid and flexible testing.

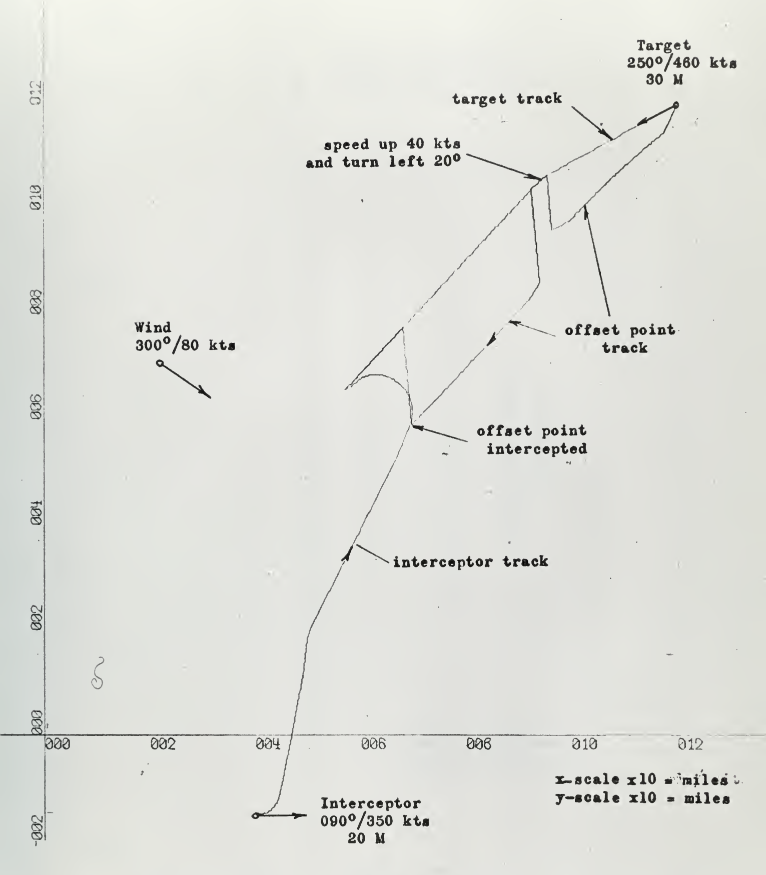



Figure 13. Idealized Intercept Solution

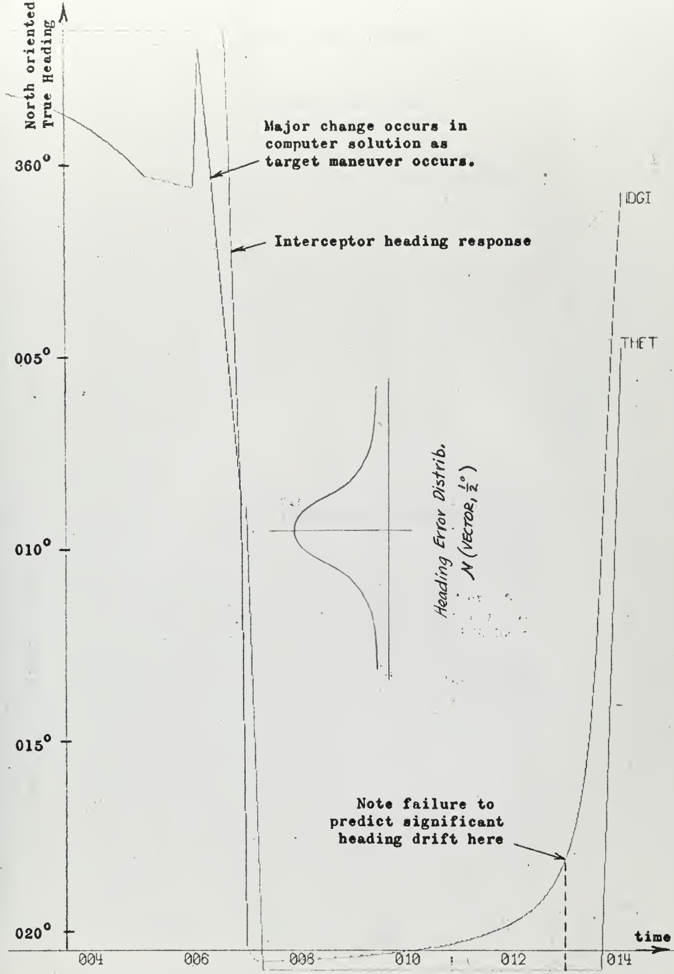



Figure 14. Heading Response with Simple Predictor

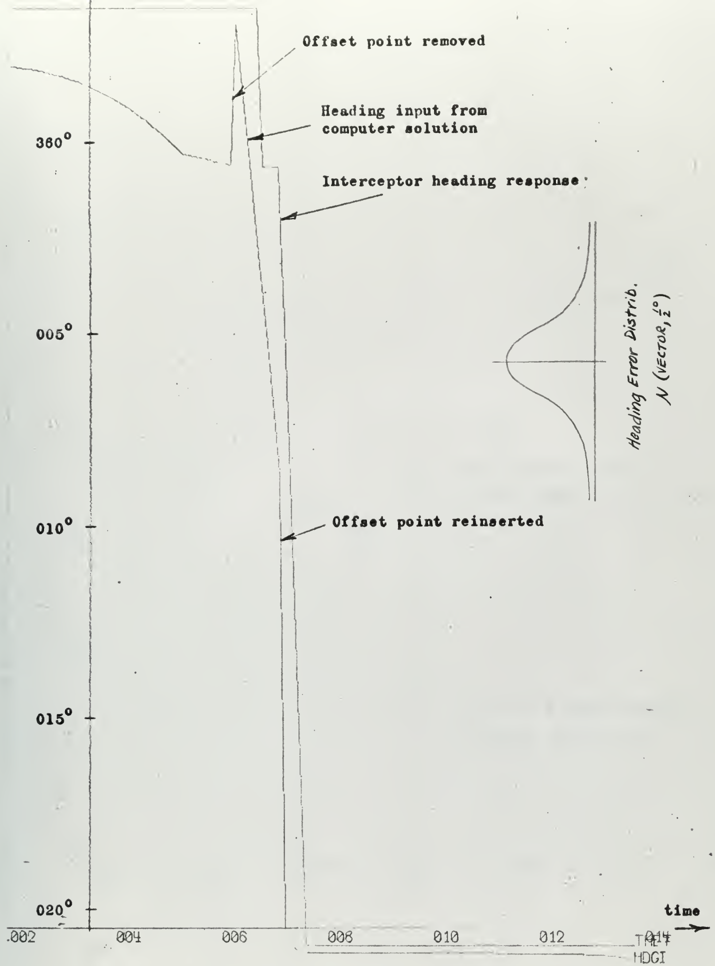



**Figure 15. Heading Response with 2nd Order Extrapolator**

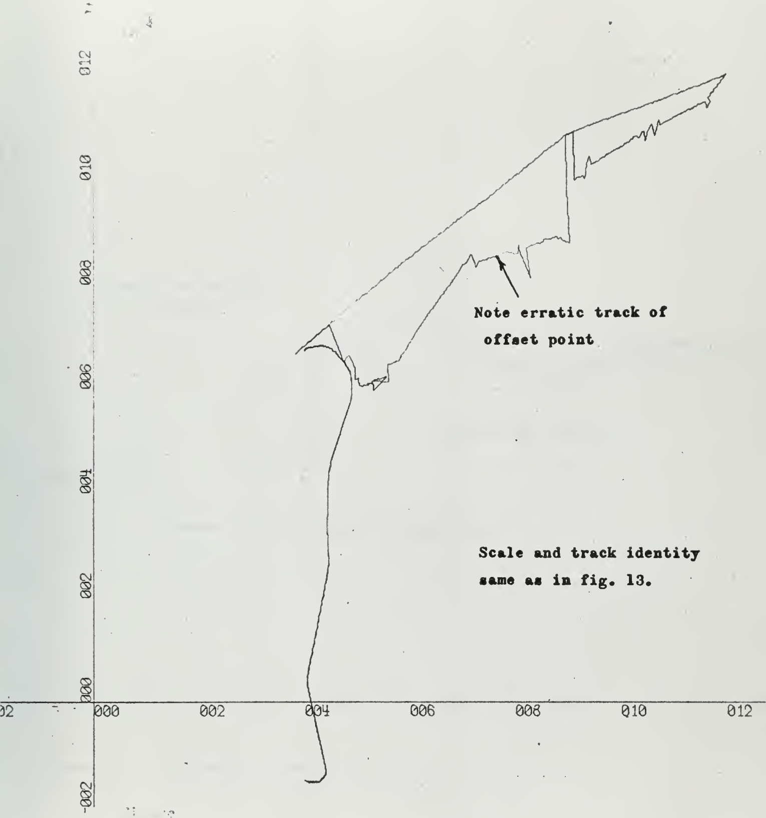



Figure 16. Intercept Solution with Variables Not Filtered

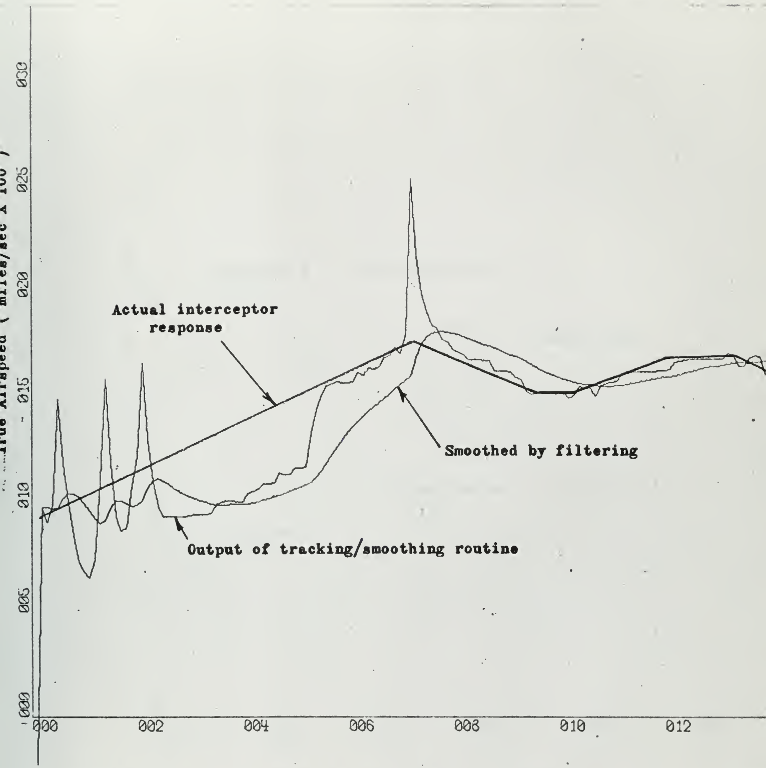



**Figure 17. Interceptor Velocity vs. Time**

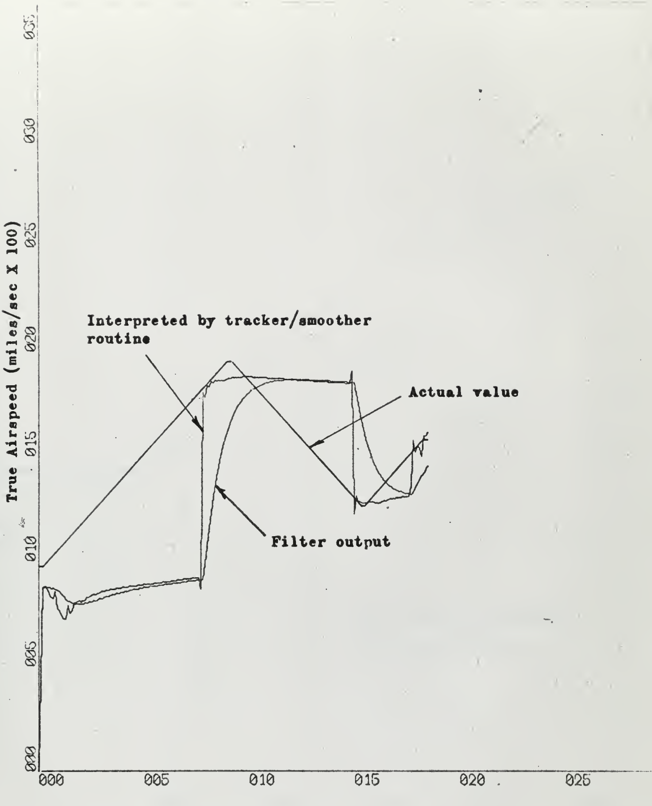



Figure 18. Interceptor Velocity vs. Time

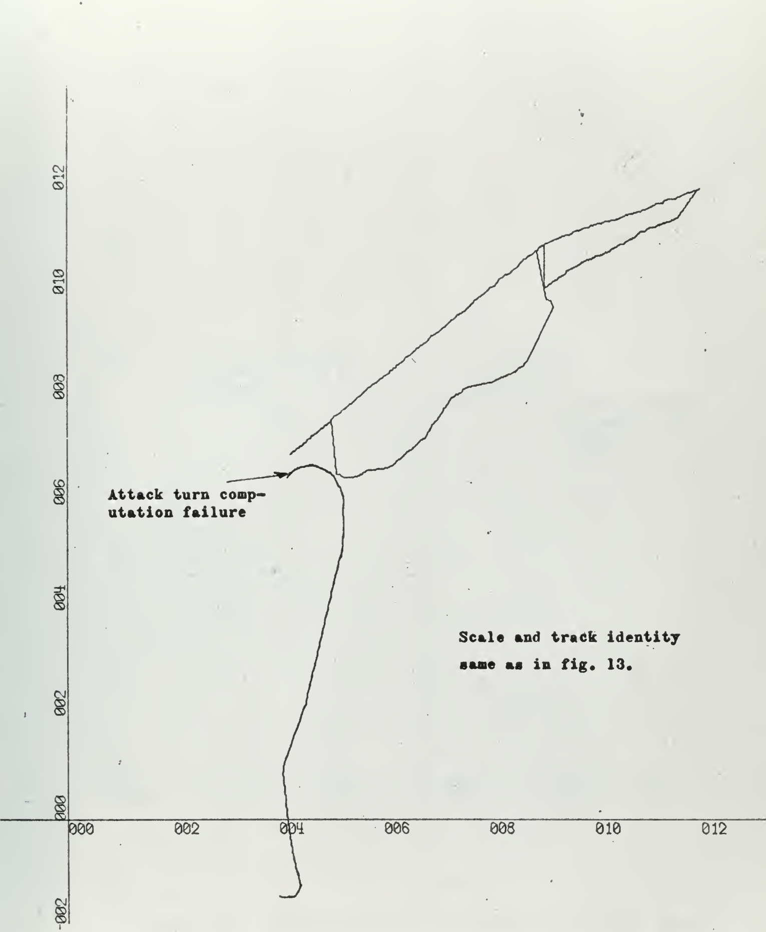



Figure 19. Intercept Solution with Filtered Variables

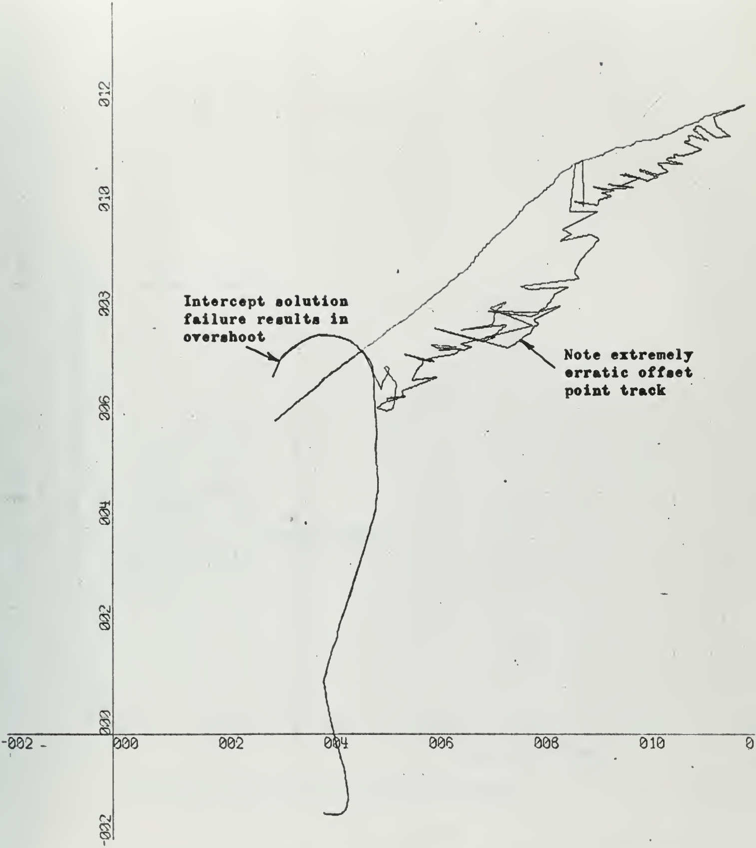



Figure 20. Intercept Solution with Filtered Variables using radar data of poor resolution.

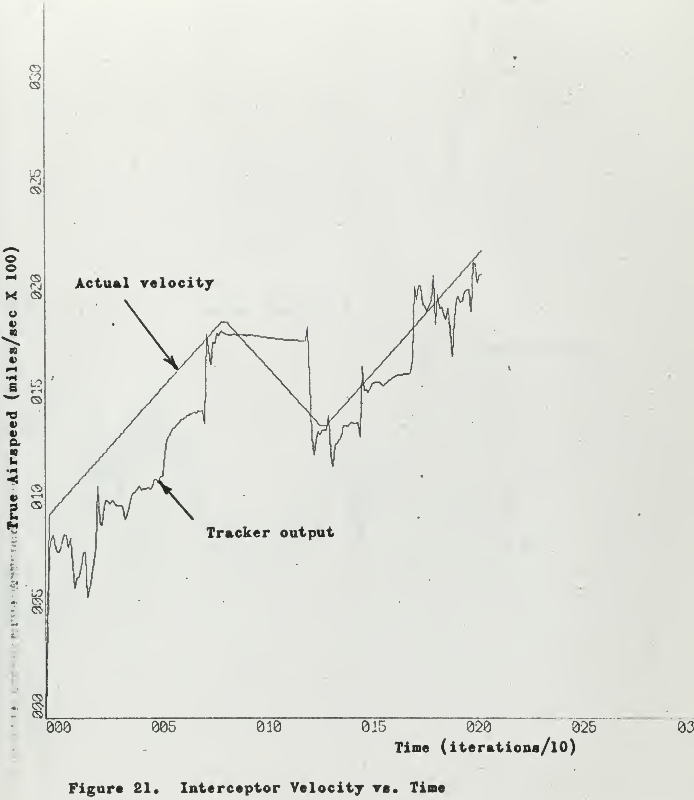



Figure 21. Interceptor Velocity vs. Time

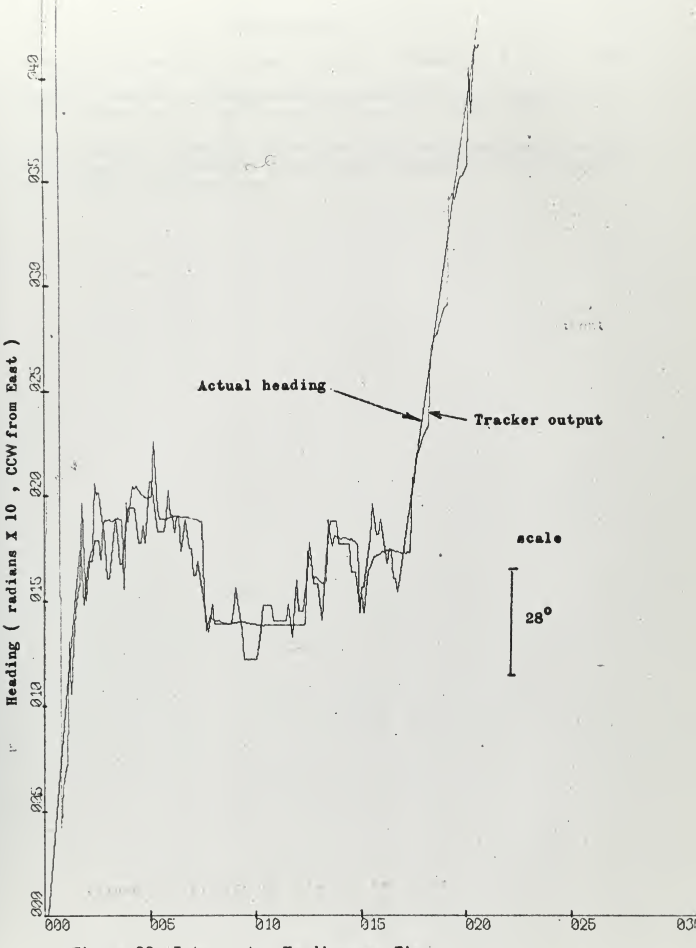



Figure 22. Interceptor Heading vs. Time

BIBLIOGRAPHY

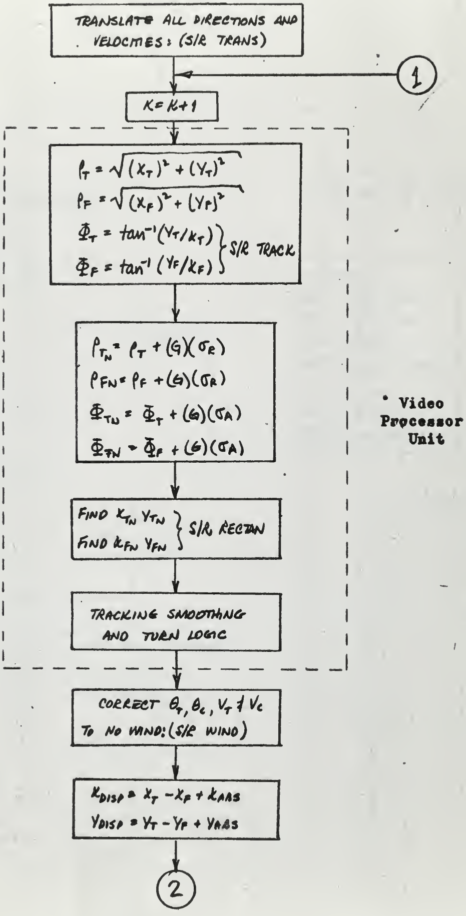
1. Gibson, J. E. Nonlinear Automatic Control, McGraw Hill Book Co., 1963.

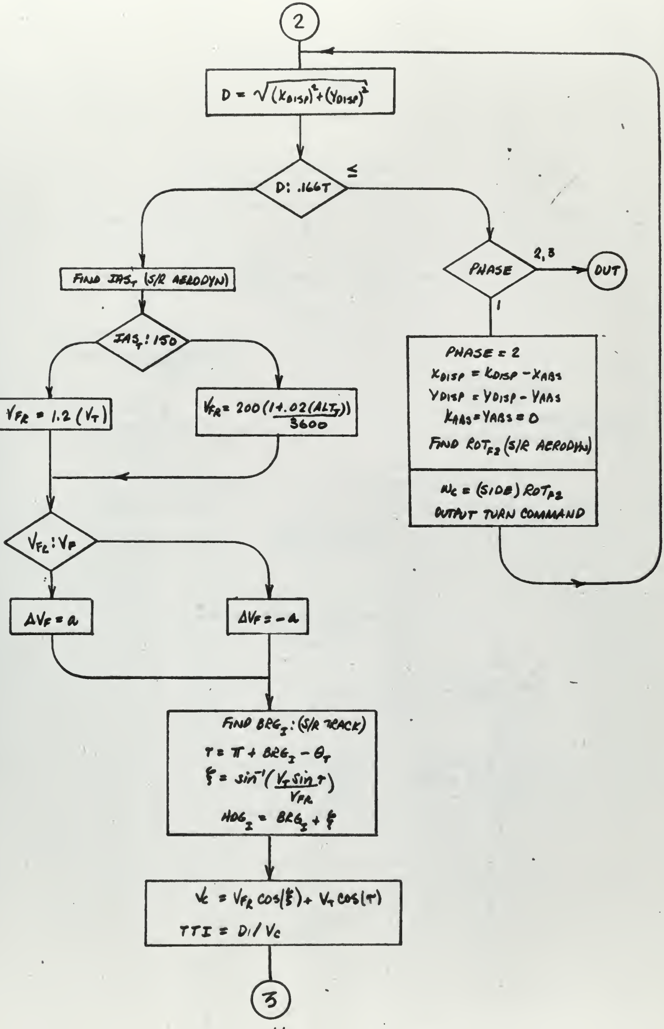
2. Boxer, R. Analysis of Sampled Data Systems and Digital Computers in the Frequency Domain, RADC technical report 55-7, April 1955.

3. Fleet Computer Programming Center, San Diego, NTDS Implementation of the Derivation of Recursive Least Squares Smoothing Factors, Informal Report 2041, 6 February 1964.

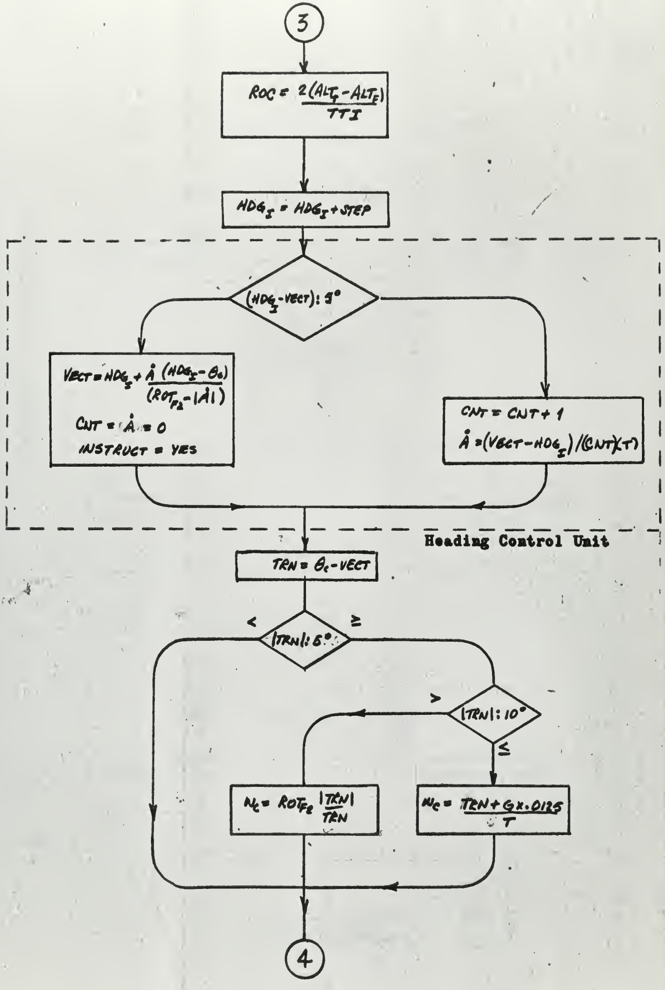
APPENDIX I

BLOCK DIAGRAM OF SIMULATOR PROGRAM

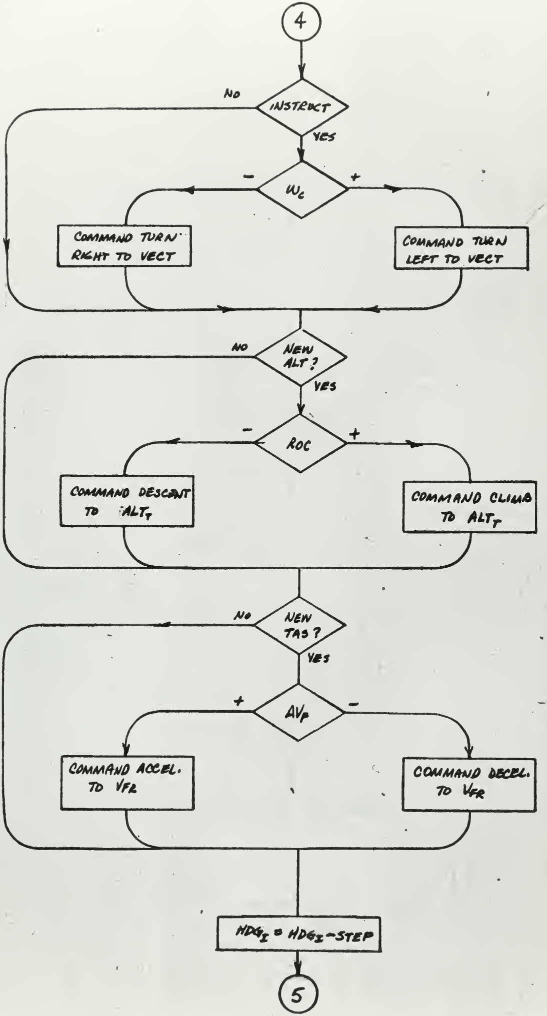


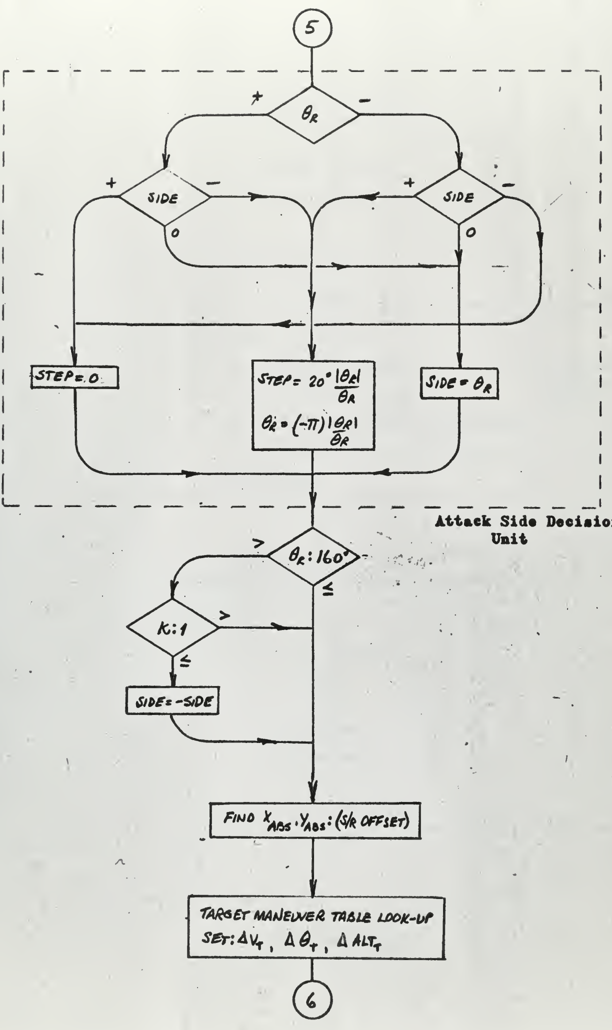


2
D = √((XDISP)² + (YDISP)²)
D: .166T
≤
FIND IAST (S/R AERODYN)
IAST: 150
VFR = 1.2 (VT)
VFR = 200(1+.02(ALTT)/3600)
VFR: VF
ΔVF = a
ΔVF = -a
FIND BRGI: (S/R TRACK)
τ = π + BRGI - θT
ξ = sin⁻¹(VT sin τ / VFR)
HDGI = BRGI + ξ
VC = VFR cos(ξ) + VT cos(τ)
TTI = DI / VC
3
PHASE
2, 3
OUT
1
PHASE = 2
XDISP = XDISP - XABS
YDISP = YDISP - YABS
XABS = YABS = 0
FIND ROTF2 (S/R AERODYN)
ωC = (SIDE) ROTF2
OUTPUT TURN COMMAND

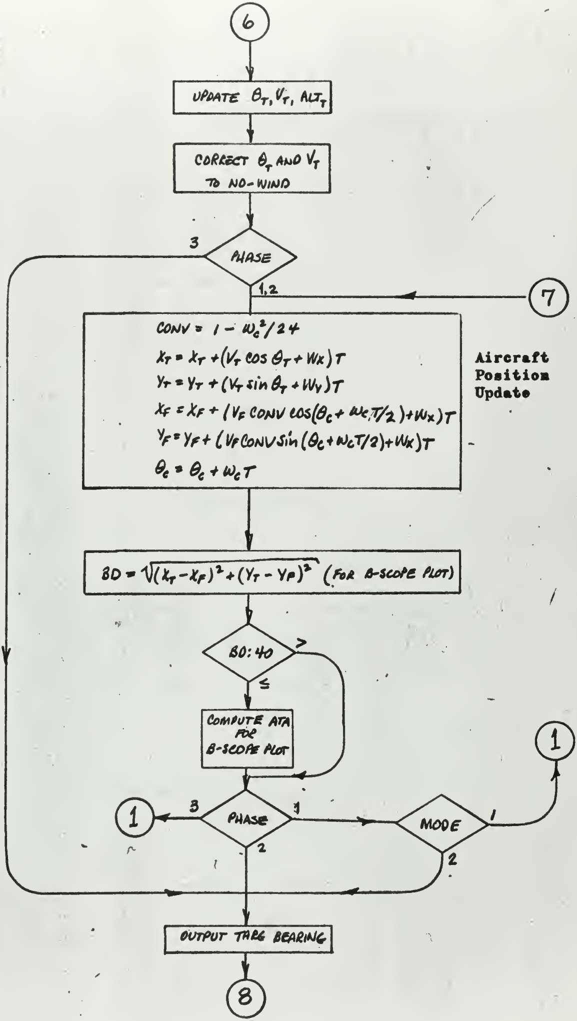


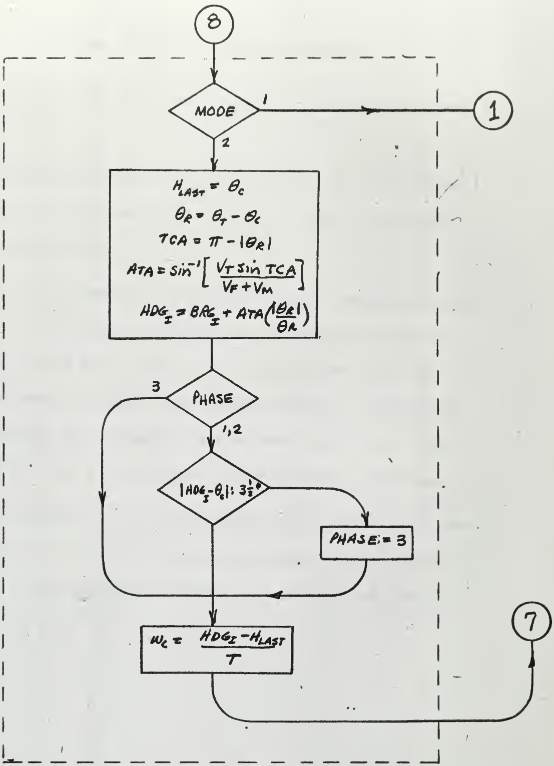
3
ROC = 2(ALT_T - ALT_E) / TTI
HDG_I = HDG_I + STEP
(HDG_I - VECT) : 5°
VECT = HDG_I + Å (HDG_I - θ_c) / (ROT_F2 - |Å|)
CNT = Å = 0
INSTRUCT = YES
CNT = CNT + 1
Å = (VECT - HDG_I) / (CNT)(T)
Heading Control Unit
TRN = θ_c - VECT
<
≥
|TRN| : 5°
>
|TRN| : 10°
≤
ω_c = ROT_F2 |TRN| / TRN
ω_c = TRN + G x .0125 / T
4


4
NO
INSTRUCT
YES
-
+
$W_c$
COMMAND TURN RIGHT TO VECT
COMMAND TURN LEFT TO VECT
NO
NEW ALT?
YES
-
+
ROC
COMMAND DESCENT TO $ALT_T$
COMMAND CLIMB TO $ALT_T$
NO
NEW TAS?
YES
+
-
$AV_F$
COMMAND ACCEL. TO $V_{FR}$
COMMAND DECEL. TO $V_{FR}$
$HDG_I = HDG_I - STEP$
5




**Attack Side Decision Unit**


6
UPDATE $\theta_T, V_T, ALT_T$
CORRECT $\theta_T$ AND $V_T$ TO NO-WIND
PHASE
3
1,2
7
$CONV = 1 - \omega_c^2/24$
$X_T = X_T + (V_T \cos\theta_T + W_X)T$
$Y_T = Y_T + (V_T \sin\theta_T + W_Y)T$
$X_F = X_F + (V_F\, CONV \cos(\theta_c + \omega_c T/2) + W_X)T$
$Y_F = Y_F + (V_F\, CONV \sin(\theta_c + \omega_c T/2) + W_X)T$
$\theta_c = \theta_c + \omega_c T$
Aircraft Position Update
$BD = \sqrt{(X_T - X_F)^2 + (Y_T - Y_F)^2}$ (FOR B-SCOPE PLOT)
BD: 40
>
≤
COMPUTE ATA FOR B-SCOPE PLOT
PHASE
3
1
2
1
MODE
1
2
1
OUTPUT TARG BEARING
8




**AI Computer Simulator**

## MORE NOTES ON SIMULATOR PROGRAM

1. MODE identifies the type run called; 1 for beam type, and 2 for head-on type.

2. PHASE indicates the phase of the intercept; 1 until offset point reached, 2 during attack turn until 3 when intercept is taken over by aircraft.

3. STEP is a $20^{\circ}$ increment to facilitate faster rotation of the bearing line during head-on intercepts, and faster off-track displacement during beam runs. It is added to HDG prior to entry into the heading control and command section, and removed before proceeding into attack side decision unit. Except for unusual positioning, it is normally zero for beam type runs.

4. Iterations begin at 1 and the time lapse between each successive value of K represents one sampling period. Actual program used a scale factor throughout to control the length of this period.

The following are the subroutines used by the simulator and referred to in the block diagram of the system by S/R.

* * * *

OFFSET subroutine computes $X_{ABS}$ and $Y_{ABS}$ from entry arguments shown:

SUBROUTINE OFFSET $(\theta_R, \theta_{TR}, RAD_F, ROT_F, V_T, X_{ABS}, Y_{ABS})$

$$D_T = \frac{V_T \, |\theta_R|}{ROT_{F2}} - .5$$

$$Y_{REL} = RAD_{F2} \left(1 - \cos(|\theta_R|)\right)$$

$$X_{REL} = D_T - RAD_{F2} \sin(|\theta_R|)$$

$$SIDE = |\theta_R| / \theta_R$$

$$X_{ABS} = X_{REL} \cos(\theta_{TR}) - SIDE \cdot Y_{REL} \sin(\theta_{TR})$$

$$Y_{ABS} = X_{REL} \sin(\theta_{TR}) + SIDE \cdot Y_{REL} \cos(\theta_{TR})$$

WIND subroutine computes true heading $T_H$ and true airspeed $T_V$ by vector subtraction of wind from relative track.

SUBROUTINE WIND $(R_H, RV, W_H, W_V, T_H, T_V)$

$$\beta = W_H - R_H$$

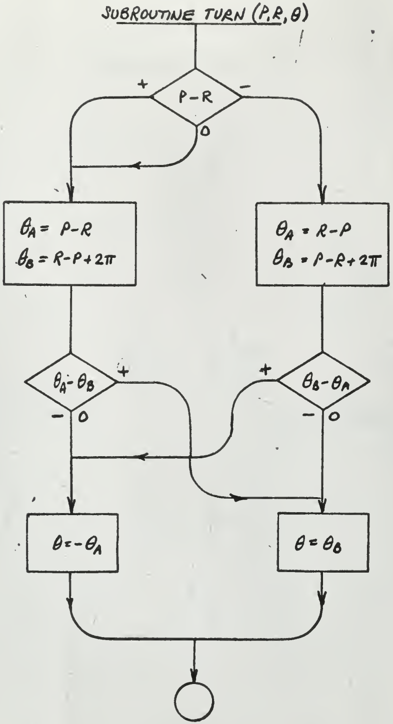
$$\gamma = RV + WV(\cos(\beta))$$

$$\rho = \tan^{-1}\left(\frac{WV \sin \beta}{\gamma}\right)$$

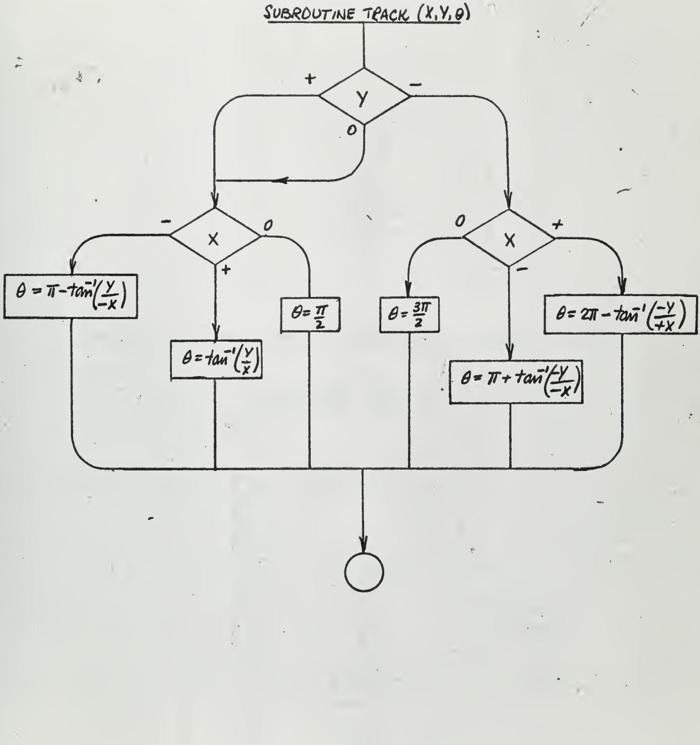
$$TH = R_H + \rho$$

$$TV = \gamma / \cos(\rho)$$

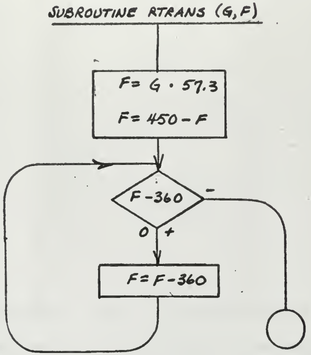
TURN subroutine finds direction and magnitude of shortest turn from P radians to R radians.



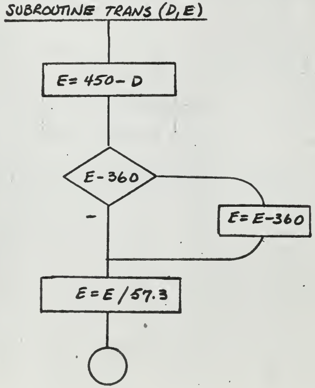
TRACK subroutine solves ARCTAN problem for correct quadrant according to the signs of X and Y arguments.



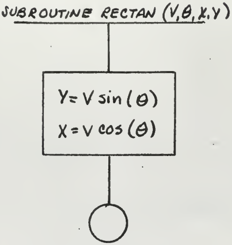
RTRANS subroutine converts radians value G to north oriented direction F in degrees.



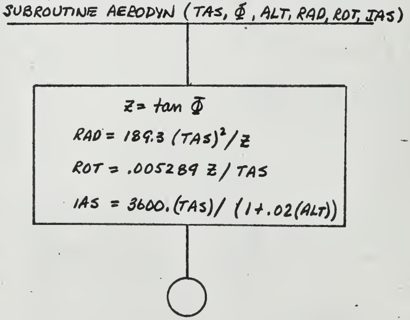

TRANS subroutine converts north-oriented heading in degrees to radians.



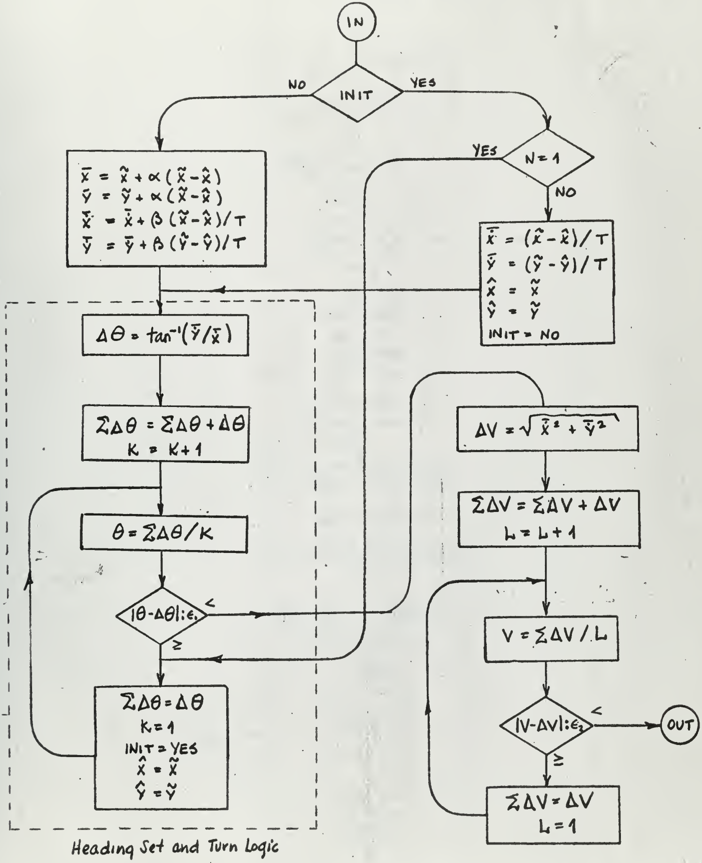
RECTAN subroutine converts polar coordinate to rectangular coordinates.




AERODYN subroutine solves for radius of turn RAD, rate of turn ROT, and indicated airspeed for given bank angle $\Phi$, altitude ALT, and true airspeed TAS.



Turn logic routine associated with constant parameter tracking method.



Turn logic used with variable parameter tracking method.